1. Consider the signal

$$
x[n]=\cos \left(\omega_{1} n\right)+a \cos \left(\omega_{2} n\right),
$$

where $a$ is a constant.
a. Write a MATLAB program that will
i. $\quad \operatorname{plot} \mathrm{x}[\mathrm{n}]$,
ii. compute the $N$ point DFT $X[k]$ (using FFT routines available within MATLAB),
iii. plot $|X[k]|$.

Turn in a printout of your M-file with your homework.
b. Run your program and generate output for the cases shown in the table below. Turn in the plots generated for each case.
c. Discuss the significance of each case.

| Case | $N$ | $\omega_{1}$ | $a$ | $\omega_{2}$ |
| :---: | :---: | :---: | :---: | :---: |
| 1 | 20 | 0.62831853 | 0.0 | - |
| 2 | 200 | 0.62831853 | 0.0 | - |
| 3 | 20 | 0.64402649 | 0.0 | - |
| 4 | 200 | 0.64402649 | 0.0 | - |
| 5 | 200 | 0.64402649 | 0.2 | 1.27234502 |
| 6 | 200 | 0.64402649 | 0.2 | 0.79168135 |

2. a. Derive a decimation-in-time FFT algorithm for an 8 point DFT, and draw a complete flow diagram for the algorithm.
b. Calculate the approximate number of complex operations required to compute the 8 point FFT and compare with the number of complex operations required to compute the 8 point DFT directly.
3. a. Derive a decimation-in-time FFT algorithm for a 12 point DFT, and draw a complete flow diagram for the algorithm.
b. Calculate the approximate number of complex operations required to compute the 12 point FFT and compare with the number of complex operations required to compute the 12 point DFT directly.
4. You have software to compute the forward DFT, but no software to compute the inverse DFT. Devise an approach to using your forward FFT software to actually compute the inverse DFT by first preprocessing the data, taking a forward DFT, then performing some postprocessing on the output from the FFT algorithm.
5. You want to exactly compute an exact 5120 -point DFT. You have a radix 2 FFT subroutine that computes the DFT for $N=2^{M}$ points for any integer value of $M$.
a. Show how to use this subroutine to efficiently compute a 5120 -point DFT.
b. Draw a block diagram for your algorithm, showing the radix 2 FFT subroutine as a black box with no detail regarding what is inside it.
c. Calculate the approximate number of complex operations required to compute the 5120 -point DFT using your efficient approach, and compare with the number of complex operations required to compute the 5120 -point DFT directly.
